**Hypothesis**

Using this dataset <https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009> , I should be able to rank the quality of red variants of the Portuguese *Vinho Verde* wine from one to ten based on features resulting from physicochemical tests using a classification algorithm.

**Cleaning the Data**

The data did not need to be cleaned heavily, all I needed to do was separate the quality rating from the original into a separate .csv file to use as my classification.

**The Data**

In the original dataset, the values were fixed acidity, volatile acid, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol, and quality (from 1-10). After cleaning, the quality was removed from the X file and added to the Y. There are about 1600 entries.



**The Algorithms**

I first tried to use KNearest because I figured it would be a good first step in classification, and I’m not familiar with how the features could relate to each other from different classifications. If there are significant differences between features, KNearest might work well. After finding that KNearest isn’t the best, coming in at about 52% accurate on a test set, I tried decision trees. I figured decision tree might work better, as it is a more in-depth classification algorithm. Considering that the data is not as easily distinguished as I thought, with very little outliers, the decision tree algorithm may be able to look at how each feature relates to each other better than KNearest. With pruning to a max depth of 11, I got an accuracy of 61.2% on the test set, which is a slight improvement from KNearest.

**Predictions and Conclusion**

I pulled data from each quality level from the data set to use as predictions. I eliminated said entries from the data set and retrained the algorithm. As I was looking for entries to use as predictions I noticed this dataset only included qualities from 3-8. It had no entries from 1-2 and 9-10. The 3 quality entry was predicted to be a 4 quality, the 4 predicted a 5, the 5 predicted a 5, the 6 predicted a 5, the 7 predicted a 7, and the 8 predicted an 8. There were some problems with this dataset that I noticed towards the end of the process, and as the deadline is rapidly approaching. As stated before, the entries did not have examples for very low or very high quality, and there seemed to be a majority of 5 quality and 6 quality entries. If I were to do this again with the same algorithms and data, I feel like limiting the classifications to just low or high quality might make the algorithm more accurate, but less useful. The data needed to have more entries representative of all qualities, instead of having a significant majority in just a couple classifications. This dataset didn’t have enough representation of all classifications, and there were too many classifications used for the amount of data that was actually in the set.

**Code:** <https://github.com/Apidra/Curtis-Machine-Learning-Final.git>